IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

Please AMEND claims 1-3, 6, 9, and 10 in accordance with the following:

1. (CURRENTLY AMENDED) A ~~storage medium for storing a~~ compiler embodied on a medium to compile a source program in an object-oriented programming language, said compiler comprising ~~the steps of~~:

if a class-type variable is contained in an execution statement to be executed in parallel or in a parallelization directive, said execution statement to be executed in parallel or in said parallelization directive included in a class, generating an instruction to call a construction instruction routine for an object of the class, before said execution statement to be executed in parallel or an execution statement to be parallelized by said parallelization directive, in order to generate said object in addition to an original object of the class; and

if a class-type variable is contained in ~~an~~ said execution statement to be executed in parallel or in ~~a~~ said parallelization directive, said execution statement to be executed in parallel or in said parallelization directive included in the class, generating ~~an~~ said instruction to call a destruction instruction routine for ~~an~~ the generated object of the class, after said execution statement to be executed in parallel or ~~an~~ said execution statement to be parallelized by said parallelization directive, in order to destruct the generated object in addition to said original object of the class.

2. (CURRENTLY AMENDED) The ~~storage medium~~ compiler according to claim 1, said compiler further comprising ~~the steps of~~:

when generating an intermediate language from said source program,

allocating a construction and destruction instruction information region in the intermediate language of the class, if a class variable which has possibility to be executed in parallel is specified; and

storing into said construction and destruction instruction information region, information concerning ~~a~~ said construction instruction routine and ~~a~~ said destruction instruction routine of an object of the class, ~~and~~

wherein information stored in said construction and destruction instruction information region is used in said ~~steps of~~ generating of ~~an~~ said instruction to call ~~a~~ said construction instruction routine and generating of ~~an~~ said instruction to call ~~a~~ said destruction instruction routine.

3. (CURRENTLY AMENDED) The ~~storage medium~~compiler according to claim 2, wherein said construction and destruction instruction information region is structured so as to be accessed from the type information of said class.

4. (ORIGINAL) The storage medium according to claim 1, wherein said compiler is a compiler for a parallel computer with shared memory.

5. (CURRENTLY AMENDED) A compiling apparatus for compiling a source program in an object-oriented programming language, comprising:

~~means for~~a first generator that generates, if a class-type variable is contained in an execution statement to be executed in parallel or in a parallelization directive, said execution statement to be executed in parallel or in said parallelization directive included in a class, generating an instruction to call a construction instruction routine for an object of the class, before said execution statement to be executed in parallel or ~~an~~ said execution statement to be parallelized by said parallelization directive, in order to generate said object in addition to an original object of the class; and

~~means for~~a second generator that generates, if a class-type variable is contained in ~~an~~ said execution statement to be executed in parallel or in ~~a~~ said parallelization directive, said execution statement to be executed in parallel or in said parallelization directive included in the class, generating ~~an~~ said instruction to call a destruction instruction routine for ~~an~~ the generated object of the class, after said execution statement to be executed in parallel or ~~an~~ said execution statement to be parallelized by said parallelization directive, in order to destruct the generated object in addition to said original object of the class.

6. (CURRENTLY AMENDED) The compiling apparatus according to claim 5, further comprising:

~~means for~~an allocator that allocates ~~allocating~~ a construction and destruction instruction information region in the intermediate language of the class during generation of an intermediate language from said source program, if a class variable which has possibility to be executed in

parallel is specified; and

~~means for~~a storing unit that stores ~~storing~~ into said construction and destruction instruction information region, information concerning a construction instruction routine and a destruction instruction routine of an object of the class, and

wherein information stored in said construction and destruction instruction information region is used by ~~said means for generating an instruction to call a construction instruction routine and said means for generating an instruction to call a destruction instruction routine~~said first and second generators.

7. (ORIGINAL) The compiling apparatus according to claim 6, wherein said construction and destruction instruction information region is structured so as to be accessed from the type information of said class.

8. (ORIGINAL) The compiling apparatus according to claim 1, wherein said compiling apparatus is a compiling apparatus for a parallel computer with shared memory.

9. (CURRENTLY AMENDED) A compiling method for compiling a source program in an object-oriented programming language, said compiling method comprising ~~the steps of~~:

if a class-type variable is contained in an execution statement to be executed in parallel or in a parallelization directive, said execution statement to be executed in parallel or said parallelization directive included in the class, generating an instruction to call a construction instruction routine for an object of the class, before said execution statement to be executed in parallel or ~~an~~ said execution statement to be parallelized by said parallelization directive, in order to generate said object in addition to an original object of the class; and

if a class-type variable is contained in ~~an~~ said execution statement to be executed in parallel or in ~~a~~ said parallelization directive, said execution statement to be executed in parallel or said parallelization directive included in the class, generating ~~an~~ said instruction to call a destruction instruction routine for ~~an~~ the generated object of the class, after said execution statement to be executed in parallel or ~~an~~ said execution statement to be parallelized by said parallelization directive, in order to destruct the generated object in addition to said original object of the class.

10. (CURRENTLY AMENDED) The compiling method according to claim 1, further comprising ~~the steps of~~:

when generating an intermediate language from said source program,

allocating a construction and destruction instruction information region in the intermediate language of the class, if a class variable which has possibility to be executed in parallel is specified; and

storing into said construction and destruction instruction information region, information concerning ~~a~~ said construction instruction routine and ~~a~~ said destruction instruction routine of an object of the class, and

wherein said information stored in said construction and destruction instruction information region is used in said ~~steps of~~ generating of ~~an~~ said instruction to call ~~a~~ said construction instruction routine and generating ~~an~~ of said instruction to call ~~a~~ said destruction instruction routine.

11. (ORIGINAL) The compiling method according to claim 10, wherein said construction and destruction instruction information region is structured so as to be accessed from the type information of said class.

12. (ORIGINAL) The compiling method according to claim 9, wherein said compiling method is a compiling method for a parallel computer with shared memory.